VS4210

HIGH DEFINITION

QUAD VIDEO PROCESSOR

Headquarters
Room B7, 1F, No.1, Li_Hsin Rd. I, Science Based Industrial Park, HsinChu City, 300, Taiwan, R.O.C.

VXIS Technology Corp. reserves the right to change or modify the information contained herein without notice.
Contact VXIS or visit the website to ensure the most recent revision of the document.

Table of Contents

1 OVERVIEW

1.1 DESCRIPTION

VS4210 is an advanced quad video processor IC. It consists of video input format converter, picture enhancement and color processing, the scaling engine for different video output size, video output format converter for varous output data format, and OSD display for video information. It receives digitized video stream from various video receiver, such as HDMI, DVI, SDI, AHD, TVI, video decoder .... The data format can be 8-bit or 16-bit, bt.656 like or bt.1120 like and resolution up to 1080P.

The VS4210 can accept four different format or different resolution videos, and combine them into one frame depends on the window foramat setting. Each video in the windows can be scaled individually. During single video display, the video can be switched to another input source seamlessly.

The VS4210 can perform high quality picture enhancement such as video noise reduction, sharpening, black-level / white-level extension, gamma correction. The output resolution covers from 320x240 to 1920x1080. The VS4210 provides frame-rate conversion, synchronization regeneration, and automatic source mode detection.

The font-based on-screen-display (OSD), and universal programmable timing control makes it become a highly integrated, most cost-efficient video processor.

1.2 APPLICATION

- Video Switcher
- Surveillance for multi camera
- Car quad monitor
- Video format converter
- Multimedia panel

1.3 FEATURES

- Support Four Digital Video Input.

  Each Video can be the following format

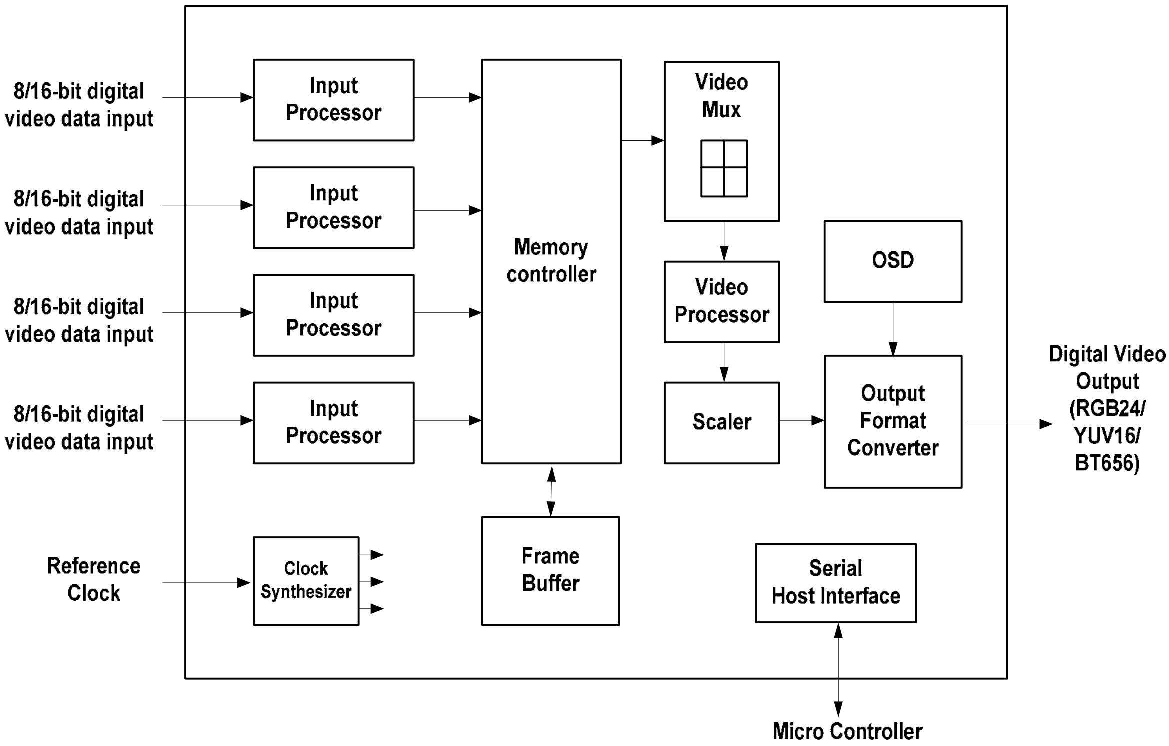
  - 8-bit bt.656/bt.601 like input data with external sync or embedded sync
  - 16-bit YUV/BT.1120 input with or without external sync
  - Resolution up to 1080P
- Support One Digital Video Output.

  The Video Output Formats can be the following format

  - 24/18/16-Bit RGB + Horizontal Sync + Vertical Sync
  - 24/18/16-Bit 4:4:4 YUV + Horizontal Sync + Vertical Sync
  - 24-Bit 4:4:4 YPbPr + Horizontal Sync + Vertical Sync
  - 16-Bit 4:2:2 YUV + Horizontal Sync + Vertical Sync
  - 8-bit YUV progressive / interlace
  - Resolution up to 1080P
- Quad Video Display. The video in each window can be selected from the four input video.
- Various window combination format for quad video display.
- Full Screen display with seamless switch
- Frame rate up-conversion to 60 Hz
- Programmer Scaling Engine, Supporting progressive Output Resolution from 320X240 to 1920X1080
- Brightness, Contrast, Saturation, and Hue Adjustment
- Color Transient Improvement, Adaptive Black-Level Extension
- Video Noise Reduction
- Frequency Directive Picture Sharpening
- Image flip, mirror, still
- 3-Channel 10-Bit Build-In Color gamma Look-Up Table for Video Fine-Tune
- Host Interface Compatible with Two-Wire IIC, Serial Interface
- OSD with 128 Build-in and 64 Programmable Font and Attribute Table, 16 Colors at same Time from 16,777,216-Color Template, Blinking, and Blending
- R/G/B output port swap & rotation control
- Y/C input port swap & rotation control
- Embedded Frame buffer
- One 27 MHz crystal
- 1.8V / 3.3V power supply
- 128pin LQFP

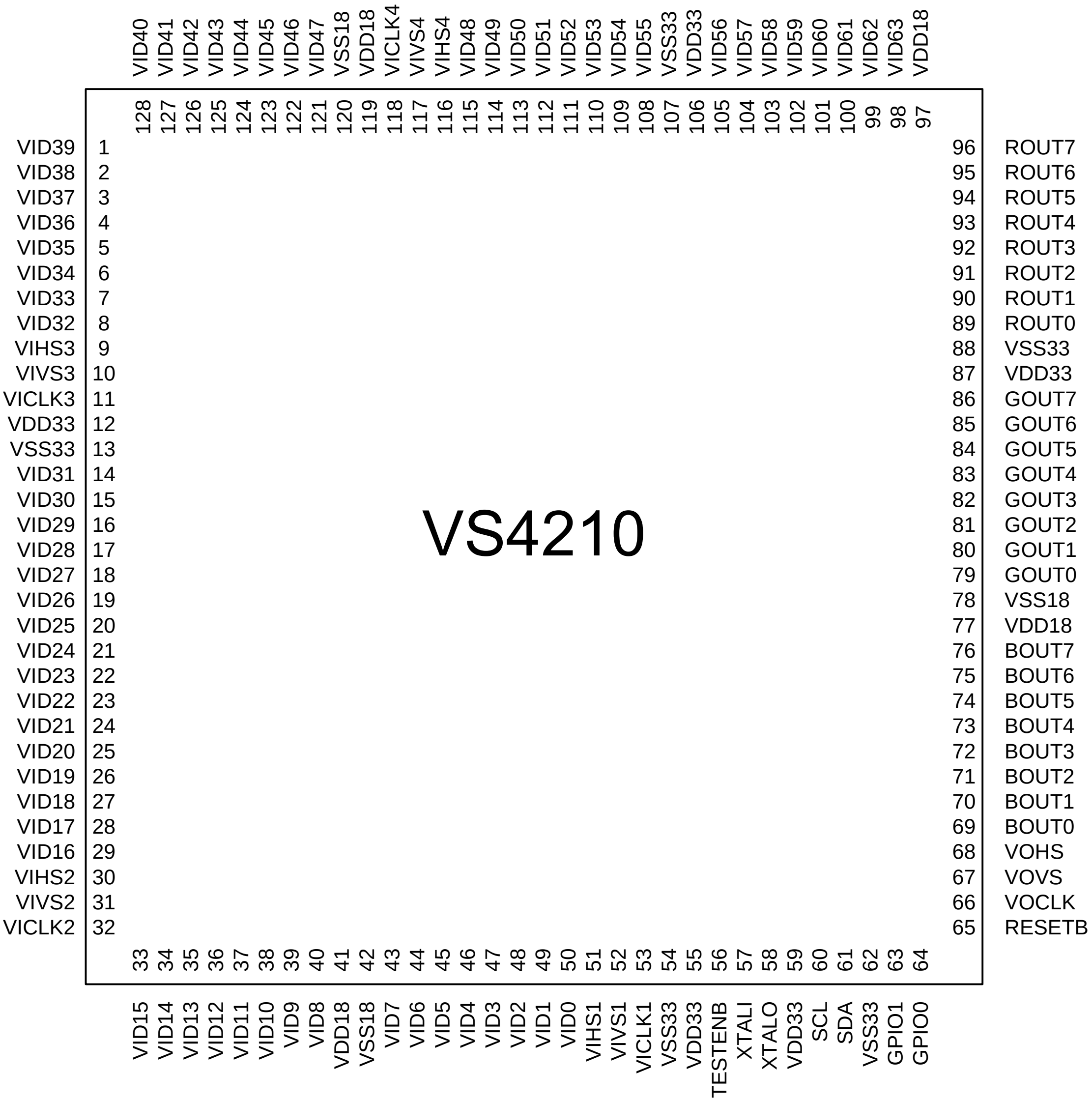
1.4 BLOCK DIAGRAM

1.4.1 BLOCK DIAGRAM OF VS4210



1.5 PINOUT DIAGRAM

1.5.1 PINOUT DIAGRAM OF VS4210



1.6 PIN ASSIGNMENT

1.6.1 PIN ASSIGNMENT OF VS4210

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
1	VID39	33	VID15	65	RESETB	97	VDD18
2	VID38	34	VID14	66	VOCLK	98	VID63
3	VID37	35	VID13	67	VOVS	99	VID62
4	VID36	36	VID12	68	VOHS	100	VID61
5	VID35	37	VID11	69	BOUT0	101	VID60
6	VID34	38	VID10	70	BOUT1	102	VID59
7	VID33	39	VID9	71	BOUT2	103	VID58
8	VID32	40	VID8	72	BOUT3	104	VID57
9	VIHS3	41	VDD18	73	BOUT4	105	VID56
10	VIVS3	42	VSS18	74	BOUT5	106	VDD33
11	VICLK3	43	VID7	75	BOUT6	107	VSS33
12	VDD33	44	VID6	76	BOUT7	108	VID55
13	VSS33	45	VID5	77	VDD18	109	VID54
14	VID31	46	VID4	78	VSS18	110	VID53
15	VID30	47	VID3	79	GOUT0	111	VID52
16	VID29	48	VID2	80	GOUT1	112	VID51
17	VID28	49	VID1	81	GOUT2	113	VID50
18	VID27	50	VID0	82	GOUT3	114	VID49
19	VID26	51	VIHS1	83	GOUT4	115	VID48
20	VID25	52	VIVS1	84	GOUT5	116	VIHS4
21	VID24	53	VICLK1	85	GOUT6	117	VIVS4
22	VID23	54	VSS33	86	GOUT7	118	VICLK4
23	VID22	55	VDD33	87	VDD33	119	VDD18
24	VID21	56	TESTENB	88	VSS33	120	VSS18
25	VID20	57	XTALIN	89	ROUT0	121	VID47
26	VID19	58	XTALOUT	90	ROUT1	122	VID46
27	VID18	59	VDD33	91	ROUT2	123	VID45
28	VID17	60	SCL	92	ROUT3	124	VID44
29	VID16	61	SDA	93	ROUT4	125	VID43
30	VIHS2	62	VSS33	94	ROUT5	126	VID42
31	VIVS2	63	GPIO1	95	ROUT6	127	VID41
32	VICLK2	64	GPIO0	96	ROUT7	128	VID40

1.7 PIN DESCRIPTION

Video Input Pins

Name	Type	Description	Notes
VID63~0	I	Digital Video Input Data	
VIHS4~1	I	Digital Video Input Horizontal Synchronization	
VIVS4~1	I	Digital Video Input Vertical Synchronization	
VICLK4~1	I	Digital Video Input Clock	

Video Output Pins

Name	Type	Description	Notes
ROUT7~0	O	R/V/Pr Digital Video Output	
GOUT7~0	O	G/Y/BT656 Digital Video Output	
BOUT 7~0	O	B/U/Pb Digital Video Output	
VOHS	O	Video Output Horizontal Synchronization	
VOVS	O	Video Output Vertical Synchronization	
VOCLK	O	Video Output Clock	
GPIO1	O	general purpose signal output 1	
	I	Host interface chip ID setting 1	
GPIO0	O	general purpose signal output 0	
	I	Host interface chip ID setting 0	

Miscellaneous I/O Pins

Name	Type	Description	Notes
/RESETB	I_{PU}	Chip Reset (Active Low)	
XTAL_OUT	XO	Crystal Output	
XTAL_IN	XI	Crystal Input	
SDA	I_{PU}/O	Host Interface Serial Data / Address	
SCL	I_{PU}	Host Interface Serial Clock	
/TEST_ENB	I_{PU}	Test Mode Enable (Active Low)	

Power Pins

Name	Type	Description	Notes
VDD33	P_{33}	Digital 3.3V power for I/O	Qty: 5
VSS33	G	Digital Ground For I/O	Qty: 5
VDD18	P_{18}	Digital 1.8V Power for Core	Qty: 4
VSS18	G	Digital Ground for Core	Qty: 3

note :

I	3.3V input
O	3.3V output
I/O	3.3V input/output
I_{PU}	3.3V input with internal pull up
XI,XO	crystal input, output pin
P_{33}	3.3V power pin
P_{18}	1.8V power pin
G	Ground pin

1.8 PACKAGE

- 128-Pin LQFP (14mm*14mm)

2 VIDEO I/O PIN ASSIGNMENT

2.1 DIGITAL VIDEO INPUT ASSIGNMENT

The VS4210 digital video interface can support 8-bit or 16-bit video data. The 8-bit video data can be bt.656 like with or without external sync. The 16-bit data can be bt.1120 like Y/C data with or without external sync.

The pin arrangement for each format is defined in *Table 2.1.*

Table 2.1 Digital Video Input Pin Assignment

Pin Name	I/O Type	Digital Input	
		8-bit YUV	16-bit YUV
VDIN63	I	YUV4[7]	Y4[7]
VDIN62	I	YUV4[6]	Y4[6]
VDIN61	I	YUV4[5]	Y4[5]
VDIN60	I	YUV4[4]	Y4[4]
VDIN59	I	YUV4[3]	Y4[3]
VDIN58	I	YUV4[2]	Y4[2]
VDIN57	I	YUV4[1]	Y4[1]
VDIN56	I	YUV4[0]	Y4[0]
VDIN55	I	-	UV4[7]
VDIN54	I	-	UV4[6]
VDIN53	I	-	UV4[5]
VDIN52	I	-	UV4[4]
VDIN51	I	-	UV4[3]
VDIN50	I	-	UV4[2]
VDIN49	I	-	UV4[1]
VDIN48	I	-	UV4[0]
VDIN47	I	YUV3[7]	Y3[7]
VDIN46	I	YUV3[6]	Y3[6]
VDIN45	I	YUV3[5]	Y3[5]
VDIN44	I	YUV3[4]	Y3[4]
VDIN43	I	YUV3[3]	Y3[3]
VDIN42	I	YUV3[2]	Y3[2]
VDIN41	I	YUV3[1]	Y3[1]
VDIN40	I	YUV3[0]	Y3[0]
VDIN39	I	-	UV3[7]
VDIN38	I	-	UV3[6]
VDIN37	I	-	UV3[5]
VDIN36	I	-	UV3[4]
VDIN35	I	-	UV3[3]
VDIN34	I	-	UV3[2]
VDIN33	I	-	UV3[1]
VDIN32	I	-	UV3[0]
VDIN31	I	YUV2[7]	Y2[7]
VDIN30	I	YUV2[6]	Y2[6]
VDIN29	I	YUV2[5]	Y2[5]

Pin Name	I/O Type	Digital Input	
		8-bit YUV	**16-bit YUV**
VDIN28	**I**	YUV2[4]	Y2[4]
VDIN27	**I**	YUV2[3]	Y2[3]
VDIN26	**I**	YUV2[2]	Y2[2]
VDIN25	**I**	YUV2[1]	Y2[1]
VDIN24	**I**	YUV2[0]	Y2[0]
VDIN23	**I**	-	UV2[7]
VDIN22	**I**	-	UV2[6]
VDIN21	**I**	-	UV2[5]
VDIN20	**I**	-	UV2[4]
VDIN19	**I**	-	UV2[3]
VDIN18	**I**	-	UV2[2]
VDIN17	**I**	-	UV2[1]
VDIN16	**I**	-	UV2[0]
VDIN15	**I**	YUV1[7]	Y1[7]
VDIN14	**I**	YUV1[6]	Y1[6]
VDIN13	**I**	YUV1[5]	Y1[5]
VDIN12	**I**	YUV1[4]	Y1[4]
VDIN11	**I**	YUV1[3]	Y1[3]
VDIN10	**I**	YUV1[2]	Y1[2]
VDIN9	**I**	YUV1[1]	Y1[1]
VDIN8	**I**	YUV1[0]	Y1[0]
VDIN7	**I**	-	UV1[7]
VDIN6	**I**	-	UV1[6]
VDIN5	**I**	-	UV1[5]
VDIN4	**I**	-	UV1[4]
VDIN3	**I**	-	UV1[3]
VDIN2	**I**	-	UV1[2]
VDIN1	**I**	-	UV1[1]
VDIN0	**I**	-	UV1[0]

For some applications or circuit design needed in system, user want to change the pin assignment different from above. The register YCIN_MUX(0x19[2:0]) can be used to swap the YC input port as *Table 2.1.2.*

YCIN_MUX	Y	C
0	Y	C
1	C	Y
7	Y	Y
2~6	Res.	Res.

Table 2.1.2 Input Y/C data port swap

Also the data pin order in each YC port can be swap inversely by register Y_INV(0x19[4]) to change the bit Y7~Y0 to Y0~Y7. And by register C_INV(0x19[3]) to change the bit C7~C0 to C0~C7.

2.2 DIGITAL VIDEO OUTPUT ASSIGNMENT

For output, VS4210 sends progressive video data out with 24/18/16-bit 4:4:4 RGB/YUV digital formats or 16-bit 4:2:2 YUV format or 8-bit YUV format through pins DR_V, DG_Y, and DB_U. The pin assignment for each format is defined in *Table 2.2*

Table 2.2 Digital Video Output Pin Assignment

Pin Name	I/O Type	Digital Output				
		8-bit YUV	16-bit 4:2:2 YUV	16-bit RGB565 / YUV	18-bit RGB666 / YUV	24-bit RGB / YUV
ROUT7	**O**	-	-	R/Pr/V [4]	R/Pr/V [5]	R/Pr/V [7]
ROUT6	**O**	-	-	R/Pr/V [3]	R/Pr/V [4]	R/Pr/V [6]
ROUT5	**O**	-	-	R/Pr/V [2]	R/Pr/V [3]	R/Pr/V [5]
ROUT4	**O**	-	-	R/Pr/V [1]	R/Pr/V [2]	R/Pr/V [4]
ROUT3	**O**	-	-	R/Pr/V [0]	R/Pr/V [1]	R/Pr/V [3]
ROUT2	**O**	-	-	-	R/Pr/V [0]	R/Pr/V [2]
ROUT1	**O**	-	-	-	-	R/Pr/V [1]
ROUT0	**O**	-	-	-	-	R/Pr/V [0]
GOUT7	**O**	YUV[7]	Y [7]	G/Y [5]	G/Y [5]	G/Y [7]
GOUT6	**O**	YUV[6]	Y [6]	G/Y [4]	G/Y [4]	G/Y [6]
GOUT5	**O**	YUV[5]	Y [5]	G/Y [3]	G/Y [3]	G/Y [5]
GOUT4	**O**	YUV[4]	Y [4]	G/Y [2]	G/Y [2]	G/Y [4]
GOUT3	**O**	YUV[3]	Y [3]	G/Y [1]	G/Y [1]	G/Y [3]
GOUT2	**O**	YUV[2]	Y [2]	G/Y [0]	G/Y [0]	G/Y [2]
GOUT1	**O**	YUV[1]	Y [1]	-	-	G/Y [1]
GOUT0	**O**	YUV[0]	Y [0]	-	-	G/Y [0]
BOUT7	**O**	-	UV [7]	B/Pb/U [4]	B/Pb/U [5]	B/Pb/U [7]
BOUT6	**O**	-	UV [6]	B/Pb/U [3]	B/Pb/U [4]	B/Pb/U [6]
BOUT5	**O**	-	UV [5]	B/Pb/U [2]	B/Pb/U [3]	B/Pb/U [5]
BOUT4	**O**	-	UV [4]	B/Pb/U [1]	B/Pb/U [2]	B/Pb/U [4]
BOUT3	**O**	-	UV [3]	B/Pb/U [0]	B/Pb/U [1]	B/Pb/U [3]
BOUT2	**O**	-	UV [2]	-	B/Pb/U [0]	B/Pb/U [2]
BOUT1	**O**	-	UV [1]	-	-	B/Pb/U [1]
BOUT0	**O**	-	UV [0]	-	-	B/Pb/U [0]

The RGB output port can be swaped for circuit application by setting register POUT_RGB_MUX(0x7F[2:0]). The swap mode for RGB port is as *Table 2.2.2.*

Table 2.2.2 Output RGB data port swap

OUT_RGB_MUX	R	G	B
0	R	G	B
1	B	G	R
2	G	R	B
3	B	R	G
4	G	B	R
5	R	B	G
6	G	G	G
7	R	R	R

Also the data pin order in each RGB port can be swap inversely by register POT_R_INV(0x7F[3]), POT_G_INV(0x7F[4]), POUT_B_INV(0x7F[5]) to change the bit 7~0 to 0~7.

3 CLOCK SYSTEM

3.1 REFERENCE CLOCK SELECTION

The VS4210 requires different clock domains for progressive processor to operate. All the clocks required for VS4210 can be generated from internal PLL. The reference input clock for PLL comes from external Crystal, the suggested crystall frequency is 27MHz.

.

The output clock frequency is generated from the PLL2 by setting the reference PLL registers.

The PLL output frequency is calculated from the foloowing equation

$$PLLOutputFrequency = PLLInputFrequency \times \frac{(PLL\_NF + 1)}{(PLL\_NR + 1)} \times \frac{1}{(PLL\_OD')}$$

Where PLL_OD’ = 1, when PLL_OD1 = 0. PLL_OD’ = 2, when PLL_OD1 = 1,
PLL_OD’ = 4, when PLL_OD1 = 2. PLL_OD’ = 8, when PLL_OD1 = 3,

Eq. 3.1 PLL frequency calculation

4 HOST INTERFACE

The VS4210 host interface uses two-wire IIC compatible interface protocol, one for clock, and one for multiplexed data/address. Input pin SCL is used for host clock input while input/output pin SDA is for multiplexed host data and address signal. It can support four different chip address defined by pin SCA[1:0] (the default value is 00). The chip write address is from 0x40 to 0x46, as defined by following.

The chip write address is 0x40,0x42,0x44,0x46, as

0	1	0	0	0	SCA[1]	SCA[0]	0

The chip read address is 0x41,0x43,0x45,0x47, as

0	1	0	0	0	SCA[1]	SCA[0]	1

The SCA pins are shared with pin {GPIO1,GPIO0}. These two pins must be pulled high or pulled low to set its chip address value during the hardware reset period (i.e. pin RESET is low).

Once chip write and read addresses are configured, the host command byte sequence can be transferred to VS4210 via the serial interface. The byte sequence consists of a CHIP ADDRESS, a REGISTER ADDRESS, followed by a number of DATA BYTES. The CHIP ADDRESS and REGISTER ADDRESS must be always provided by the host, usually a micro-controller, and the DATA BYTES are provided by host for host-writings and by VS4210 for host-readings.

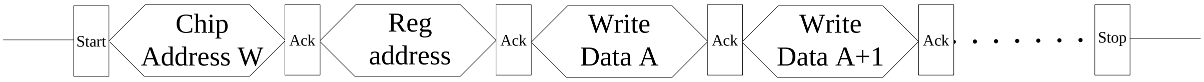


Fig 4.1 Host command write sequence

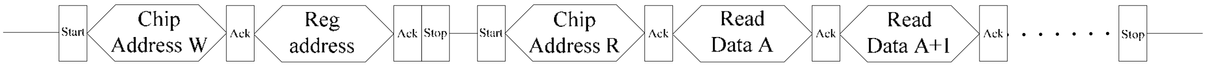


Fig 4.2 Host command read sequence

As shown above, the first DATA BYTE following REGISTER ADDRESS A is read from or write to register address A, the DATA BYTE behind will be read from or write to register address A+1, and so on. So for large host-writings such as chip initialization, only the initial register address needs to be specified once to complete whole host writing operations. In this manner user can save a lot of host command cycles in complicated applications.

Because there are four channel data for four windows to be processed in VS4210, and they share the same register control address. We use a 2-bit register CH_IIC_SET(address 0x02[1:0]) to control which window data is processed. The address of these sharing registers are from address 0x10 to 0x30, and 0xdc~0xef. When CH_IIC_SET is set to 0, then all the sharing registers are set to window 1. When CH_IIC_SET is set to 1, then all the sharing registers are set to window 2. When CH_IIC_SET is set to 2, then all the sharing registers are set to window 3. When CH_IIC_SET is set to 3, then all the sharing registers are set to window 4. And when register ALL_CH_SET(0x02[2]) is set to 1, then these sharing registers are set to all four windows simultaneously.

5 HARDWARE AND SOFTWARE RESET

VS4210 can be reset to initial status in two ways. One is through the hardware pin, RESETB; by asserting RESETB pin to ground voltage, entire chip will be reset to its initial states. The other way is through the host interface by writing to register GBL1 (0x01). Writing value 0x5A to GBL1, called software reset, will generate an internal reset pulse signal similar to RESETB to initialize the entire chip.

Similar to writing 0x5A to GBL1, writing 0xA5 to GBL1 will reset entire chip *except* control registers programmed by host interface. The use of 0xA5 writing often occurs after initial register programming or mid-state register changes to assure the chip working from the initial state.

The register ACLK_ON(0x03[0]) is used to control all the internal clock. When this bit is set to 0, then all the clocks are off for saving power.

6 VIDEO INPUT SELECTION

The VS4210 supports 8-bits or 16-bits video input. The 8-bits data can be bt656/bt601 like format with embedded or external sync. The 16-bits data can be bt1120 like with embedded or external sync. These data format are shown in Fig. 6.1, Fig. 6.2. Fig. 6.3.

The data format can be set by register INFMT[2:0] (reg 0x11[2:0]). Following Table 6.1 shows VS4210's acceptable digital video input formats. For their pin assignment, please review *CHAPTER 2 DIGITAL VIDEO I/O PIN ASSIGNMENT*.

Table 6.1 Digital Video Input Format Selection

INFMT [2:0]	Digital Video Input Format
010	16-bit Y/UV 4:2:2 data
011	8-Bit ITU-R BT.656 / BT.601
others	reserved

The VS4210 can auto detect the input sync pin VIHSx,VIVSx. If there are signals detected on these two pins, the input processor auto use these two pins for hsync and vsync, and the status register SYNC_DET(0x1B[1]) will be set to 1. If no signal detected on these two pins, the input processor will use the embedded sync inside the data. If both external and embedded sync are not detected, then no input data will be processed and the status register NO_VIDEO(register 0x1B[0]) will be set to 1. The input processor can also detect the input sync polarity to latch the active data, user can manually define the input sync polarity by setting the register MSYNC_EN(0x11[3]) to 1. Once this bit is set to 1, the user can define the input sync polarity by setting register VIHSP(0x11[6]) for hsync polarity, VIVSP(0x11[5]) for vsync polarity, and VIFSP for field polarity(0x11[4]) if input video is interlace.

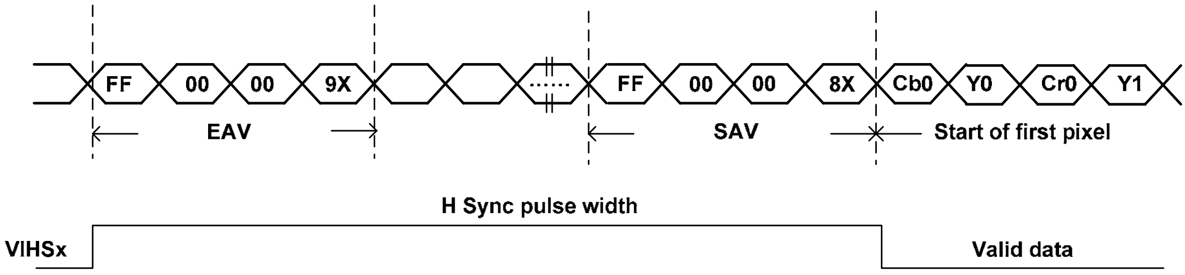


Fig 6.1 8-bit input data without/with sync format

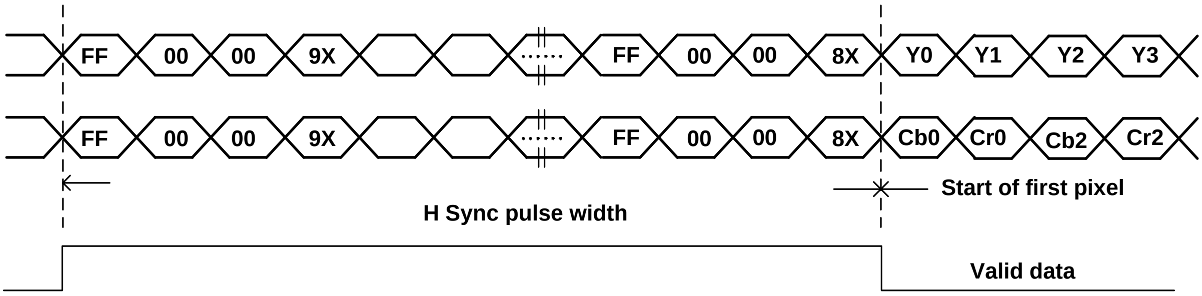


Fig 6.2 16-bit input data without/with sync format

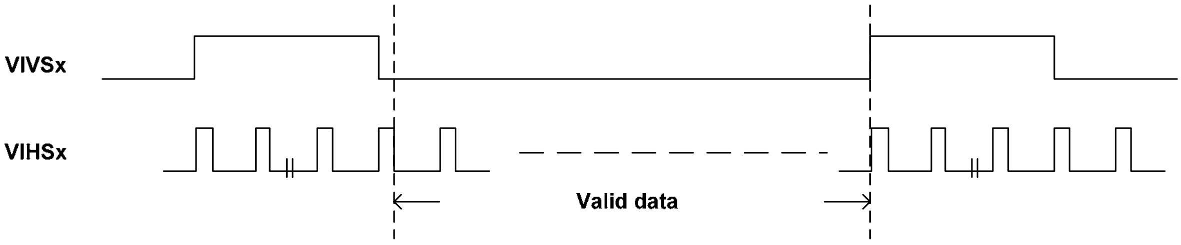


Fig 6.3 Valid Data Respect to Vsync

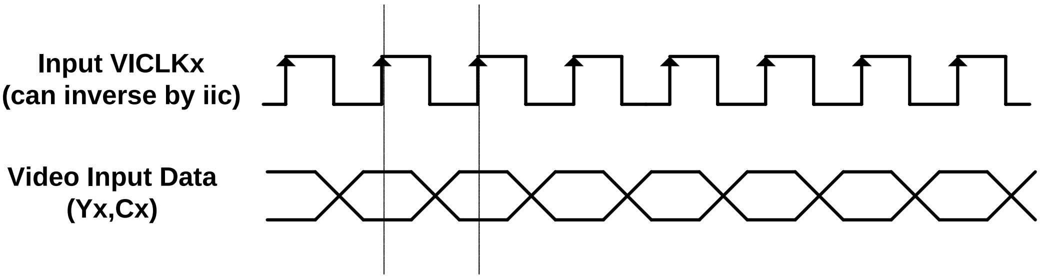


Fig 6.4 Input Data Latch

The VS4210 can also auto detect the input signal is interlace or progressive. When the input data is detected as interlace, the register INTERLACE(0x1B[2]) will be set to 1, otherwise will be set to 0 for progressive input.

The width and length and frame rate of input video can also be automatic detected and the detected values will be shown on the registers. The register HSIN_LENGTH(0x22[7:4],0x23) shows the input width of one horizontal line, the register HSIN_ACT(0x22[3:0],0x24) shows the input active width of one horizontal line. The register VSIN_LENGTH(0x25[7:4],0x26) show the input vertical line number in one frame, register VSIN_ACT(0x25[3:0],0x27) shows the input vertical active line number in one frame. The register FRAME_CNT(0xDD,0xDE,0xDF) shows the reference value to calculate frame rate of input video. The frame rate can be calculated by the formula

Frame rate = 27M / (FRAME_CNT * 10)

All the input signals and data are latched in the rising edge of input clock. To correctly detected the above input signal, the input clock phase should be adjusted to latch these data. Input clock phase can be adjusted by the input clock polarity VICLKP(0x11[7]) to change the input clock phase 180 degree, or by input clock phase shift VICLK_DLY(0x18[7:5]).

If the user want to manuallly define the input width, length or active width, active length, then the register MIL_EN(0x11[7]) can be set to 1. Once this bit is set to 1, the user must define the frame width by setting the register M_HLEN(0x1F[7:4],0x20), frame length by setting register M_VLEN(0x1F[3:0],0x21), and active width by setting M_HACT(0x12[7:4],0x13), active length by setting M_VACT(0x12[3:0],0x14), and the sync is from external or embedded data by setting register EXTSYNC_EN(0x1E[5]), and the input format is interlace or progressive by setting register M_INTERLACE(0x1A[6]).

Usually the input data is latched per cycle of the video input clock. For some cases that the one pixel data should be latched every two cycles of the video input clock, such as BT.656 or other 8-bit data input. Then the register VICLKF(0x1A[5]) should be set to 0 to get the correct data.

7 PRE-SCALER

The pre-scaler do the scaling down on both vertical and horizontal direction for the input video.
The horizontal scaling down is set by the register PRES_HSF (address 0x2D), the value 0 means no scaling in the horizontal direction. The larger the PRES_HSF value, the smaller the horizontal size. When the value equal to 0x80, the horizontal size is one half of the original size. If you want to scale down the horizontal szie more than 1/2 of the original size, set the value of PRE_H_SKIP(address 0x2B[3:0]), the horisontal size will directly shrink to 1/(PRE_H_SKIP+1) of original size.

The vertical scaling down is set by the register PRES_VSF (address 0x2C), the value 0 means no scaling in the vertical direction. The larger the PRES_VSF value, the smaller the vetrical size. When the value equal to 0x80, the vertical size is one half of the original size. If you want to scale down the vertical size more than 1/2 of the original size, set the value of PRES_V_SKIP(address 0x2B[7:4]), the vertical size will directly shrink to 1/(PRES_V_SKIP+1) of original size.

The active horizontal and vertical size after scaling down can be checked by reading the register PRES_HACT(address 0x2E[3:0], 0x2F), PRES_VACT(address 0x2E[6:4],0x30).

8 RELÁCS

Relács is VXIS's intellectual invention of high-quality and low-cost scaler. It utilizes separate algorithms for up-scaling and down-scaling processes and sharing the usage of line buffers.

The output frame-rate is arbitrary, depending on the output resolution and the output clock rate. For example, we want a 720P (active 1280 x 720) output with frame-rate 60 Hz, the total size is (1650 x 750). Then the output clock frequency should be 74.25M MHz. The general equation of above calculation is as follows. For output clock setting please refer to Chapter 3.

OutputClockFrequency = HorizontalTotalPixels x VerticalTotalLines x OutputFrameRate

Up Scaling Equation

$$VERTICAL\_SF = \frac{LCNT\_IN}{LCNT\_OUT} * 32768$$

Down Scaling Equation

$$VERTICAL\_SF = (\frac{LCNT\_IN}{LCNT\_OUT} - 1) * 32768$$

Equation 8.1 Output Clock Frequency and Scaling factor Calculation

Note that Relács keeps same frame-rate before and after scaling. To convert the output video frame-rate with respect to the input video, see next chapter for details.

9 FRAME RATE CONVERSION

Programming frame rate conversion in VS4210 is simple and intuitive. The output frame rate is dependent on the output clock and the output horizontal and vertical size as in Equation 9.1, and it is independent of the input frame rate. The maximum output frame rate of VS4210 can achieve is limited by the output clock frequency.

$$\text{Video Output Frame Rate} = \frac{\text{Output Clock Frequency}}{\text{H\_TOTAL} * \text{V\_TOTAL}}$$

Equation 9.1 Video Output Frame-Rate Calculation

10 WINDOW CONTROLLER

There are maximum four windows to be processed simultaneously in VS4210. The Windows can be arranged to variant combinations for diffent applications. To set the window combination type is by setting register WIN_MODE (0x9B[6:3]), The supported window format is as Figure 10.1.

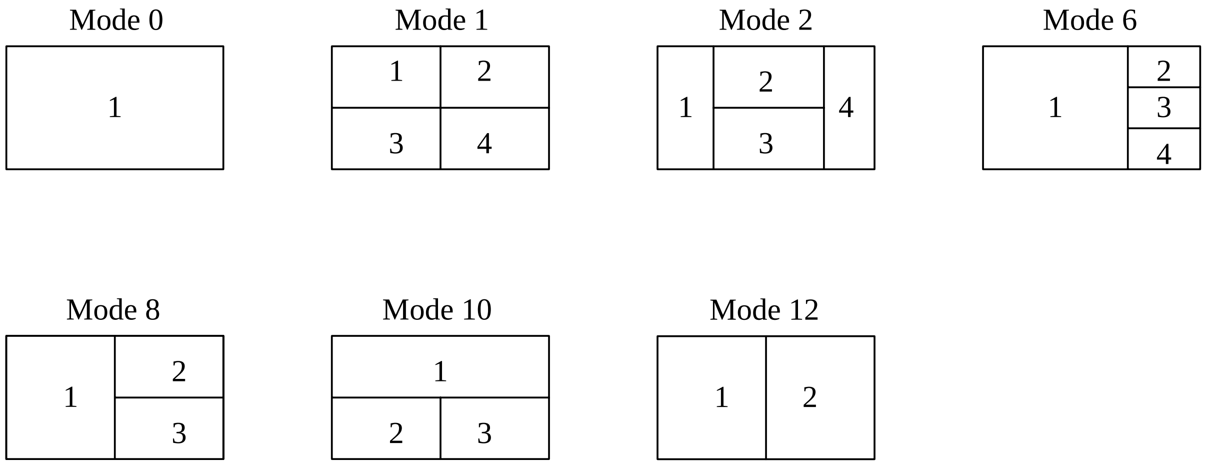


Figure 10.1 Quad Window modes

11 COLOR ENHANCEMENT

11.1 CONTRAST, BRIGHTNESS, COLOR, AND HUE ADJUSTMENT

The VS4210 supports essential color adjustment through the registers, CONTRAST, BRIGHTNESS, COLOR, and HUE. Among those, the CONTRAST and the BRIGHTNESS can also be adjusted independently in R / G / B manner by the registers, R_CONTRAST, G_CONTRAST, B_CONTRAST, R_BRIGHTNESS, G_BRIGHTNESS, and B_BRIGHTNESS.

11.2 BLACK-LEVEL EXTENSION (BLE)

Basic idea of black level extension is to enhance the contrast of the luminance in the dark potion of the picture. As the result, the average luminance in the dark potion will be extended to darker level non-linearly while the luminance in bright potion remains unchanged. The advantage of this function is to make the object more solid, apparent, and noticeable to the viewers. The BLE function works adaptively, depending on the average luminance of the picture. The Luminance transform function for BLE is controlled by three parameters, BKXLVL(0x36), BKXMAX(0x37), and BKXSLP(0x39)

11.3 VIDEO NOISE REDUCTION (VNR)

The VS4210 contains video noise reduction (VNR) engine specifically removing Gaussian noise and mid-band interception noise, which commonly occurs in video transmission channels. *Figure 11.1* shows the block diagram of VS4210's VNR engine. VNR function is enabled from the register VNR.

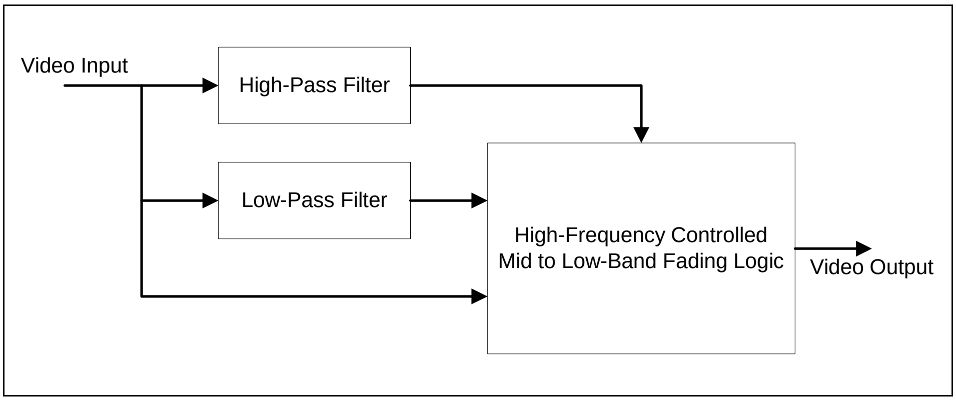


Figure 11.1 VS4210 VNR Engine Block Diagram

11.4 SHARPNESS

The VS4210 offers three peaking filters for different frequency response in horizontal sharpness engine. The gain for each filter is adjustable from 0 to 14 dB and individually controlled with registers, PEAK_ADJ1(0x4C[5:0]), PEAK_ADJ2(0x4D[5:0]), and PEAK_ADJ3(0x4E[5:0]). For peaking filter 1, 2, and 3, each respectively amplifies 1/2, 1/4, and 1/6 of sampling frequency (27 MHz). *Figure 11.2* illustrates the frequency response for each filter. Following the three peaking filters is a clipping filter to suppress video gain after peaking filters and reduce noise. The clipping filter is adjustable with the registers PEAK_CLIP_MIN(0x4B[6:0]) and PEAK_CLIP_MAX(0x4F[6:0]). And the peaking function is enabled by setting register PEAK_EN(0x4B[7]) to 1.

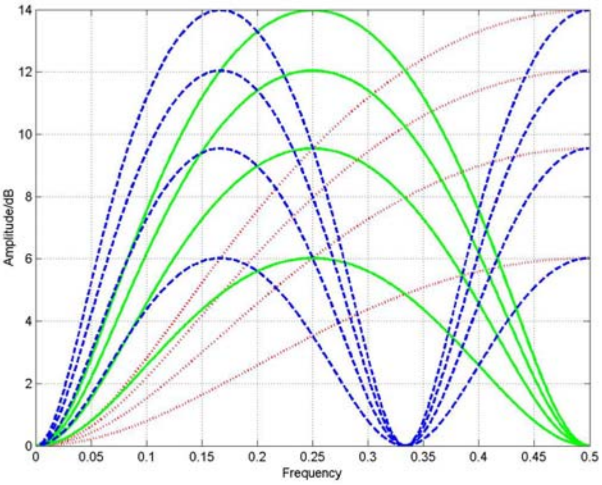


Figure 11.2 Frequency Response for Peaking Filters

11.5 COLOR-TRANSIENT IMPROVEMENT (CTI)

For most video solutions, video content in chrominance domain are often with less care due to human eye’s neglect of color variations. Therefore, we sometimes found that the picture is implicitly dirty and foggy especially when we observing the video patterns or color-bars. The color-transient improvement (CTI) engine in VS4210 is made to recover this imperfectness of video presentation and perform sharp and keen edges for every objects and overall clearer video to the viewers. The CTI function is enabled by setting register CTI_C(0x35[2]) to 1, to enable the effect on color. Or by setting register CTI_Y(0x34[6]) to 1, to enable the effect on luminance.

11.6 COLOR LOOK-UP TABLE (CLUT) AND GAMMA CORRECTION

The VS4210 provides method to finely manipulate video sequence, color look-up table (CLUT). CLUT is made of three embedded SRAMs and requires initial SRAM programming with continuous-write scheme of host interface. Register CLUT_MODE enable the usage of CLUT.

11.6.1 CLUT

Widely used in many applications, the VS4210 equips a build-in 3-channel 10-bit color look-up table (CLUT). The CLUT contains three 256 x 10-bit SRAMs, one for each of R/Pr, G/Y, and B/Pb channels. For each channel, the SRAM acts as an one-to-one mapping array and reads 8-bit data as index, and maps into 10-bit data as SRAM data output. By manipulating every single cell of SRAMs, user can make neat changes of the color mapping for each single level of 256 color levels in each channels. Using the CLUT, all the picture/video related adjustments such as color adjustment, gamma, and color temperature can be accomplished with specifically programmed mapping data. CLUT_MODE and the datawidth register CLUT_WIDTH, must be set before using the CLUT.

Programming the CLUT is intuitive through the continuous-write registers. First, program the destination SRAM to one or all of the R/Pr, G/Y, B/Pb channels with the register CW_DEST. Next, write the initial 8-bit address to register CW_INIT_ADDR. Then, in sequence write the data into the register CW_DATA, and the internal host interface controller will automatically write this data into the designated SRAM after a short period of time, and increment the address by 1 after each internal

SRAM writing. If the CLUT data width is set to 10-bit, then the internal automatic SRAM writing operation will take place at every two host writing of CW_DATA (*Figure 11.3*).

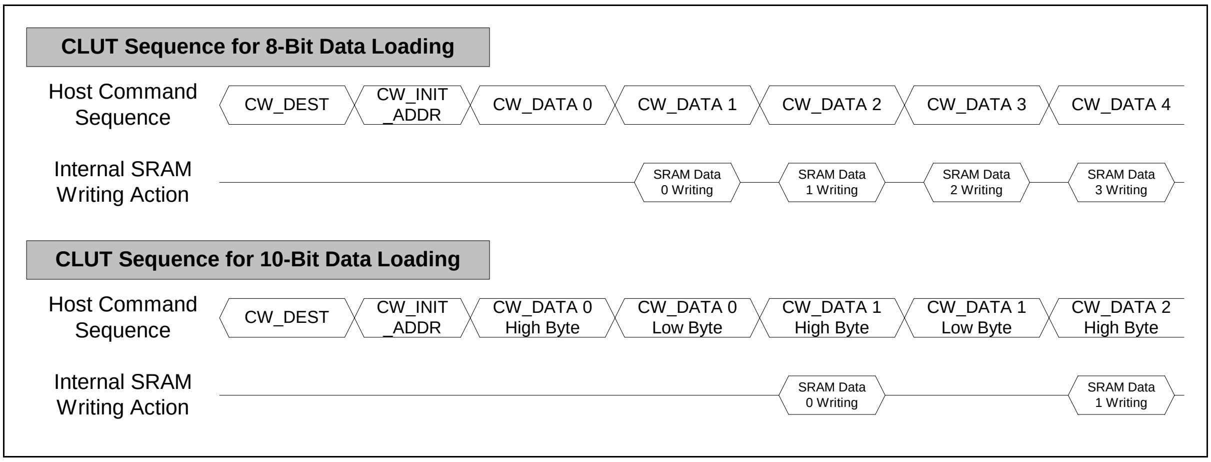


Figure 11.3 CLUT Programming Sequence Through Host Interface

Note that the CLUT always maps the video in the color domain of analog output port. This means when the analog output port is set to RGB, CLUT mapping is in RGB domain, and when the analog output is set to YPbPr or YUV, CLUT mapping is in YPbPr or YUV domain. Also, as shown in *Figure 11.8* that the value stored in the SRAM can be in 10-bit range (CLUT_WIDTH = 1), or in 8-bit range (CLUT_WIDTH = 0) in RGB or YUV domain mappings. But note that due to synchronization insertion to Y channel, in YPbPr domain CLUT only supports 8-bit mappings.

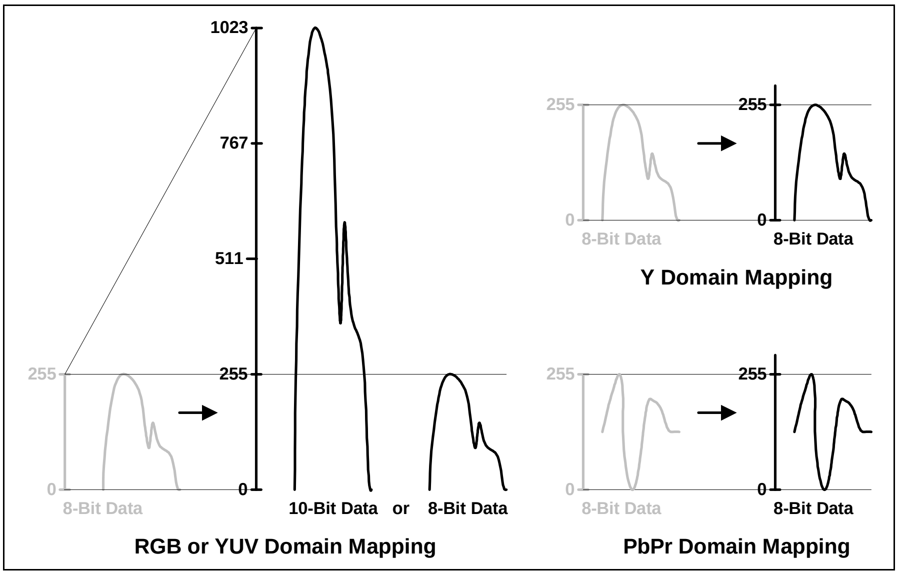


Figure 11.4 CLUT Mapping

12 VIDEO OUTPUT SECTION

12.1 PROGRESSIVE VIDEO OUTPUT SETTING

The VS4210 generates variety of digital video outputs formats. It supports digital output in 24/18/16-bit RGB, 4:4:4 YPbPr (with embedded horizontal and vertical synchronization on Y), 4:4:4 YUV (with additional HSYNC and VSYNC pins), or 16-bit 4:2:2 YUV formats. For digital video I/O pin assignment, please see "Chapter 2 *DIGITAL VIDEO I/O PIN ASSIGNMENT*". Format selection of progressive digital output ports is done through setting of register OFMT.

Table 12.1 Progressive Video Output Format

	OFMT	**Output Format**
Digital Port	0	3-Channel RGB (df.)
	1	3-Channel 4:4:4 YPbPr
	2	2-Channel 4:2:2 YUV (with additional HSYNC and VSYNC pins)
	3	3-Channel 4:4:4 YUV (with additional HSYNC and VSYNC pins)

12.2 DITHERING

When the output format is determined, there is a dithering option provided in 4:4:4 RGB / YPbPr / YUV digital video output formats. For users who are limited to accept 24/18/16-bit digital videos, two approaches can be selected, either simply truncate the 24-bit output data by connecting partial bits of digital video output, or dither the 24-bit output data by setting register DITHER (*Table 12.2*). The dithering algorithm used in the VS4210 is by applying error diffusion calculation to the data and will effectively reduce the boundary effect in low-resolution displays.

Table 12.2 Dithering Options

DITHER	**Dithering Option (G/Y : B/Pb/U : R/Pr/V)**
0	Disable
1	Dithering to 18-bits (6-bits : 6-bits : 6-bits)
2	Dithering to 24-bits (8-bits : 8-bits : 8-bits)
3	Dithering to 16-bits (6-bits : 5-bits : 5-bits)

12.3 CLAMPING

The VS4210 has an option to set the digital video output level in ITU-R BT.601 range, ITU-R BT.656 range, or simply in 0-255 8-bit full range. For ITU-R BT.601 standard, luminance channel (Y) is in the range of 16-235, and chrominance channel (UV) is in the range of 16-240. For ITU-R BT.656 standard, luminance (Y) and chrominance channel (UV) are in the range of 1-254. See the register CLAMP for details.

12.4 VIDEO OUTPUT TIMING

For digital video output, by default the digital video output data is valid at the raising edge of the video output clock VOCLK. For the device that latches the digital video data at the falling edge of the video clock, the VOCLK can be inverted by setting the register VOCLKP.

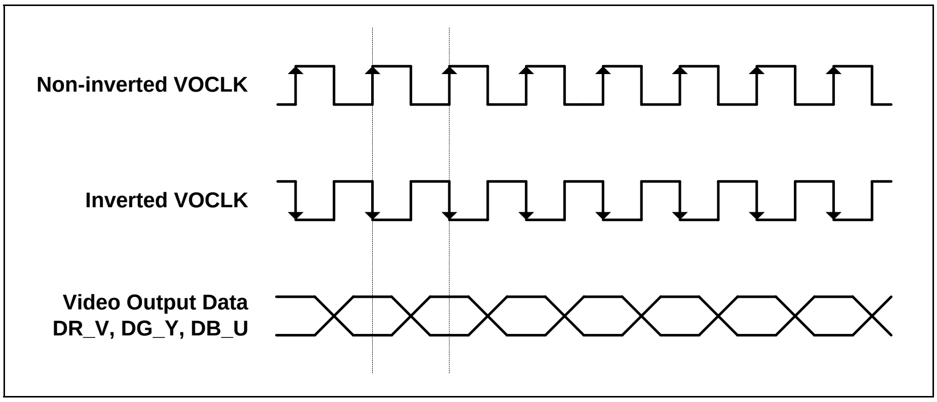


Figure 12.4 Video Output Timing

12.5 VIDEO OUTPUT SYNCHRONIZATIONS

All the video output synchronization signals, HSYNC, VSYNC, or GCSYNC, of VS4210 are adjustable with respect to output format, frequency, and resolutions. Basically the horizontal synchronization (HSYNC) asserts on every horizontal line’s boundary; vertical synchronization (VSYNC) asserts on every frame’s boundary; and general composite synchronization (GCSYNC) acts as the combination of horizontal and vertical synchronizations (*Figure 12.5.1*).

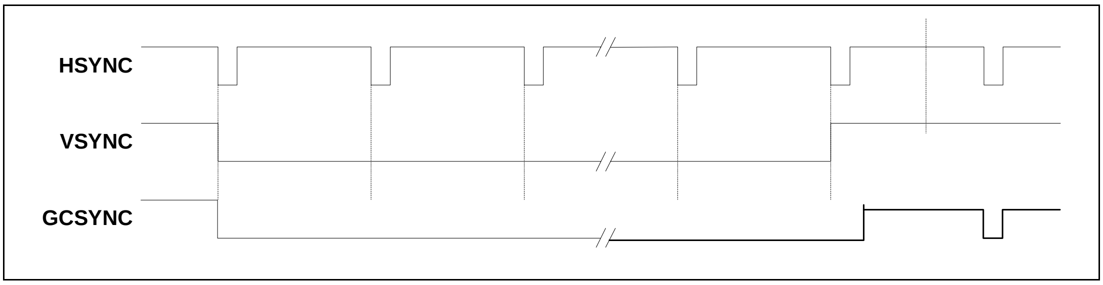


Figure 12.5.1 Video Output Synchronizations

The adjustment of synchronization-width of HSYNC and VSYNC is through the registers HS_WIDTH and VS_WIDTH.

HSYNC and VSYNC are always available at pins. GCSYNC is shared with HSYNC and switched by the register VOHS_SEL

In YPbPr video output format, the synchronization information is added to the luminance channel (*Figure 12.5.2*). The synchronization depth is adjustable through the register YSYNC_LVL.

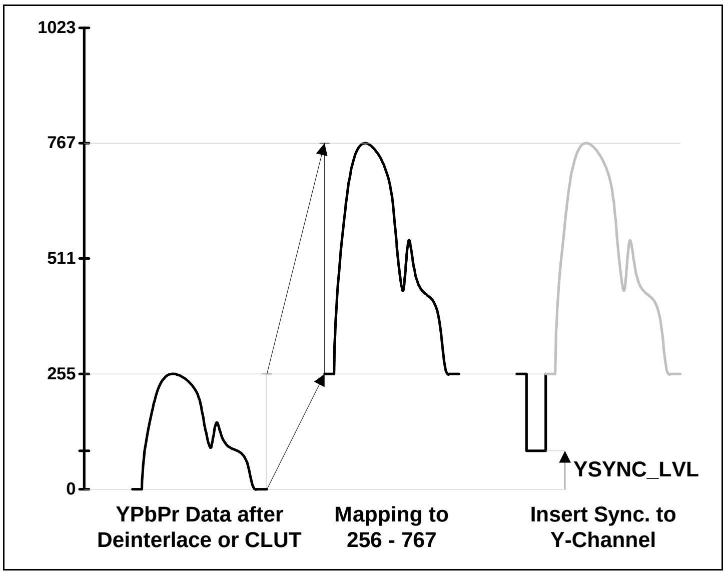


Figure 12.5.2 Synchronization on YPbPr

12.6 VIDEO OUTPUT DISPLAY

Two essential elements, screen shifting and masking, control the VS4210 video output display. The screen shifting function is activated by changing the horizontal shifting register HSHIFT, and vertical shifting register VSHIFT. Setting larger value of HSHIFT moves the picture rightward; and setting larger value of VSHIFT moves the picture downward. Also, the screen masking function is activated by setting the registers BOTTOM_MASK, TOP_MASK, LEFT_MASK, and RIGHT_MASK. The VS4210's output display region with its synchronization and control parameters (screen shifting and screen masking) is summerized in *Figure 12.6*.

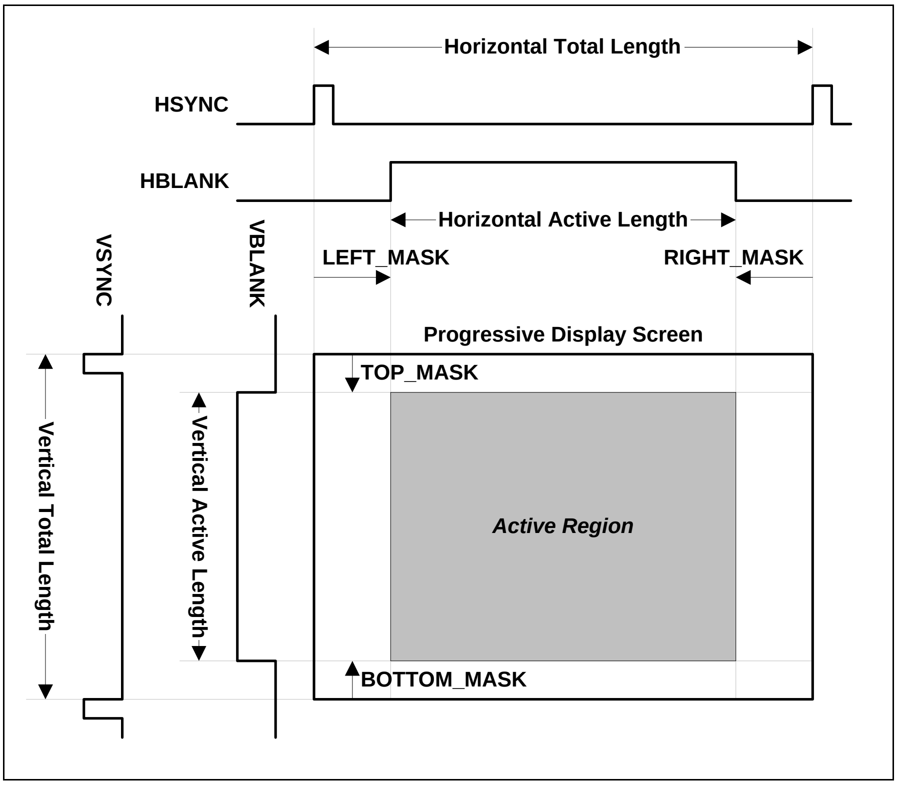


Figure 12.6 Output Display Region

12.7 OUTPUT CROSS LINE DRAWING

You can draw vertical or horizontal lines in the VS4210 output. It supports total eight lines drawing, each can be set to horizontal or vertical direction line.

Since all the line setting share the same regisers, you must set which line is selected first. The line is selected by register LINE_SEL(0xB8[2:0]). Then we can enable or disable this line drawing output by setting LINE_OSD_EN(0xB9[7]). The line width is set by register LINE_WIDTH(0xB9[3:0]), and color is set by register LINE_COLOR(0xB9[7:5]). There are eight colors can be selected for the lines, which is as table12.7

LINE_COLOR	0	1	2	3	4	5	6	7
COLOR	white	yellow	cyan	green	magenta	red	blue	black

If you want to draw horizontal line, set the register H_DIR(0xBD[7]) to value 0. The register LINE_PA({0xBA[7:4],0xBB}), LINE_PB({0xBA[3:0],0xBC}) define the start and end coordinates in the X-axis. LINE_PC({0xBD[3:0],0xBE}) defines the line coordinate in the Y_axis. If you want to draw vertical line, set the register H_DIR(0xBD[7]) to value 1. The register LINE_PA({0xBA[7:4],0xBB}), LINE_PB({0xBA[3:0],0xBC}) define the start and end coordinates in the Y-axis. LINE_PC({0xBD[3:0],0xBE}) defines the line coordinate in the X-axis.

13 ON-SCREEN-DISPLAY (OSD)

13.1 OSD INTRODUCTIONS

The VS4210 integrates VXIS's font-attribute-based on-screen display (OSD) unit, which can display a total of up to 256 characters in a single screen, with each font in 16 pixels x 20 pixels format. The embedded font Random-Access-Memory (RAM) and Read-Only-Memory (ROM) let user select characters from up to 192 fonts, 128 build-in and 64 programmable fonts. The attribute bits programming let user designate arbitrary spectacular menu, closed caption, even games from 16,777,216 colors, blinking, *Italic* font, underline font, and many artistic features.

13.2 OSD DISPLAY BLOCKS

The build-in OSD system divides the screen display into three basic sections, the title, content, and bottom blocks, and the user can customize the size and position for each display block by host commands (*Figure 13.2.1*).

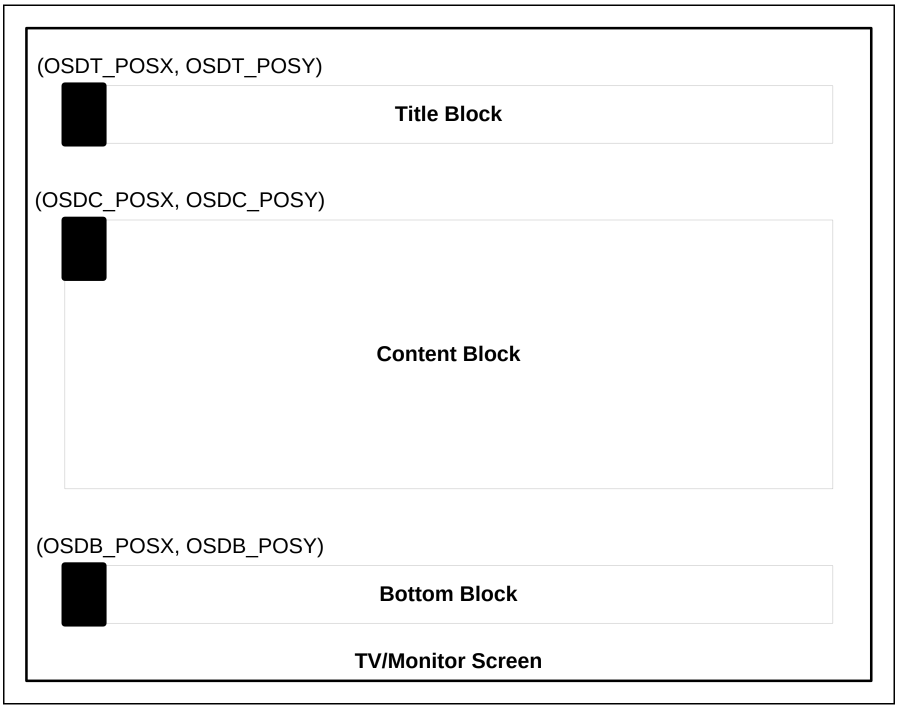


Figure 13.2.1 OSD Display Blocks

The title and bottom blocks are restricted to display one line of text commonly for header or page notes; and the content block displays multiple lines of text for main OSD information. The sizes and the positions for each individual block are adjustable through registers (*Table 13.2.1*). Each displaying block cannot be overlapped with others. For details of the register setting, check OSD section in register description chapter.

Table 13.2.1 Position and Size Registers

Register		**Description**
OSDT_POSX	OSDT_POSY	Position registers for title block
OSDC_POSX	OSDC_POSY	Position registers for content block
OSDB_POSX	OSDB_POSY	Position registers for bottom block
OSDT_SIZEX	N/A	Size registers for title block
OSDC_SIZEX	OSDC_SIZEY	Size registers for content block
OSDB_SIZEX	N/A	Size registers for bottom block

13.3 OSD OPERATIONS

The VS4210's OSD unit is font-based entry. All information that is going to be shown in the screen must be translated into fonts, which is in 16 pixel x 20 pixel resolution each, then put into the screen.

There are two types of OSD memories embedded in VS4210. One is called the "font memory", which stores all the fonts currently being used on the screen. The font memory consists of one 128-font ROM (2560 x 16-bit) for commonly used fixed fonts, and one 64-fonts RAM (1280 x 16-bit) for user-programmable fonts (*Table 13.3.1*). If utilizing maximum amount of the memory, there are up to 192 fonts can be repeatly shown in the single page.

Table 13.3.1 Font Memory Table

Index (Hex)	00h	01h	02h	03h	04h	05h	06h	07h	08h	09h	0Ah	0Bh	0Ch	0Dh	0Eh	0Fh
Character	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Index (Hex)	10h	11h	12h	13h	14h	15h	16h	17h	18h	19h	1Ah	1Bh	1Ch	1Dh	1Eh	1Fh
Character	Q	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f
Index (Hex)	20h	21h	22h	23h	24h	25h	26h	27h	28h	29h	2Ah	2Bh	2Ch	2Dh	2Eh	2Fh
Character	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
Index (Hex)	30h	31h	32h	33h	34h	35h	36h	37h	38h	39h	3Ah	3Bh	3Ch	3Dh	3Eh	3Fh
Character	w	x	y	z	á	à	â	ç	é	è	ê	í	î	Ñ	ñ	ó
Index (Hex)	40h	41h	42h	43h	44h	45h	46h	47h	48h	49h	4Ah	4Bh	4Ch	4Dh	4Eh	4Fh
Character	ô	ú	û	!	,	.	;	:	'	"	#	%	&	@	/	(
Index (Hex)	50h	51h	52h	53h	54h	55h	56h	57h	58h	59h	5Ah	5Bh	5Ch	5Dh	5Eh	5Fh
Character	)	[	]	+	−	÷	<	=	>	?	°	¢	$	£	®	™
Index (Hex)	60h	61h	62h	63h	64h	65h	66h	67h	68h	69h	6Ah	6Bh	6Ch	6Dh	6Eh	6Fh
Character	½	¿	0	1	2	3	4	5	6	7	8	9	♪			█
Index (Hex)	70h	71h	72h	73h	74h	75h	76h	77h	78h	79h	7Ah	7Bh	7Ch	7Dh	7Eh	7Fh
Character	▭	▏	▎	▍	▌	▋	▊	▉	█	^	∨	‾	_	▕	▏	□
Index (Hex)	80h	81h	82h	83h	84h	85h	86h	87h	88h	89h	8Ah	8Bh	8Ch	8Dh	8Eh	8Fh
Character	User-Programmable Fonts															
Index (Hex)	90h	91h	92h	93h	94h	95h	96h	97h	98h	99h	9Ah	9Bh	9Ch	9Dh	9Eh	9Fh
Character	User-Programmable Fonts															
Index (Hex)	A0h	A1h	A2h	A3h	A4h	A5h	A6h	A7h	A8h	A9h	AAh	ABh	ACh	ADh	AEh	AFh
Character	User-Programmable Fonts															

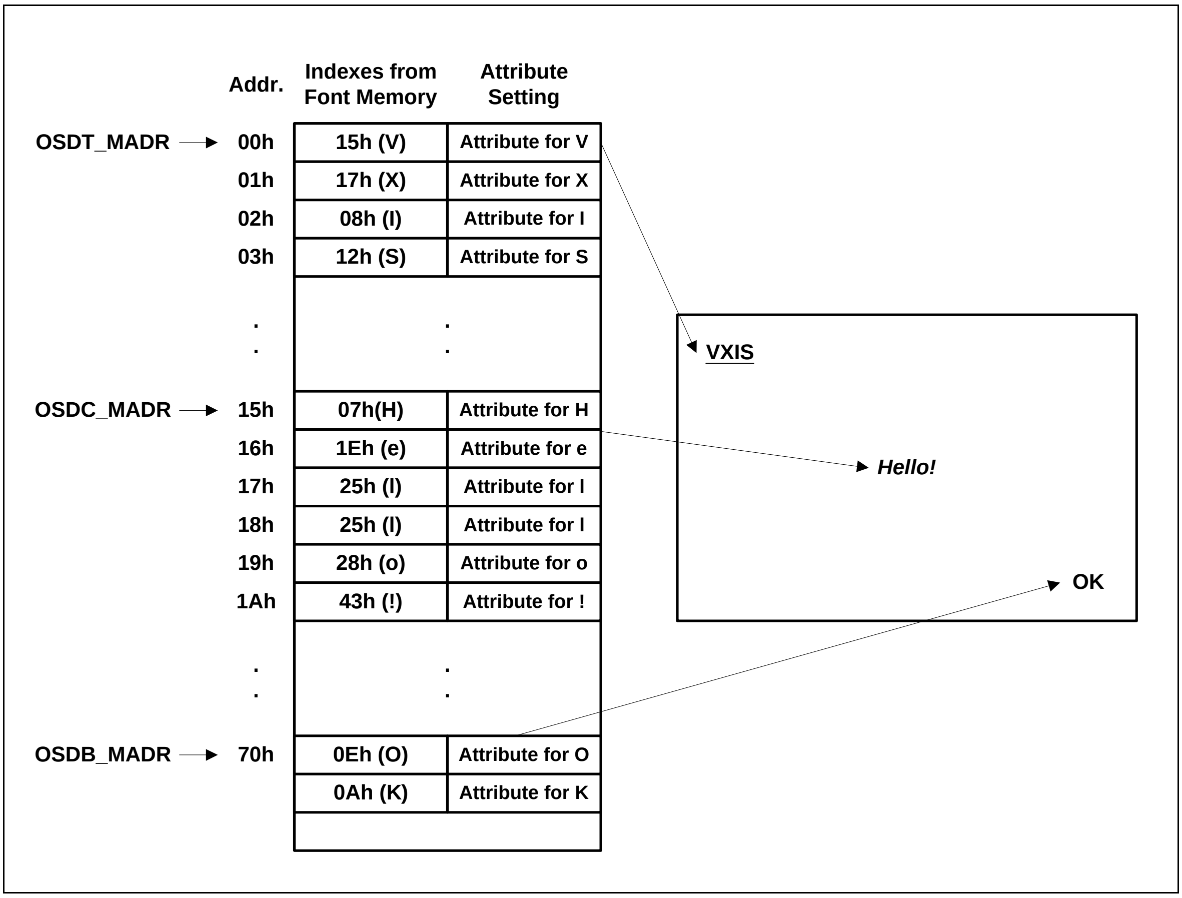
Index (Hex)	B0h	B1h	B2h	B3h	B4h	B5h	B6h	B7h	B8h	B9h	BAh	BBh	BCh	BDh	BEh	BFh
Character	User-Programmable Fonts															
Index (Hex)	C0h	C1h	C2h	C3h	C4h	C5h	C6h	C7h	C8h							
Character	CR	2B	3B	4B	5B	6B	7B	8B	9B							
CR: Character Return / Line Feed nB: Number of Space Characters																

Another type of memory is called the “command memory”, which stores the sequence and the attribute of the font that is appearing on the screen. The command memory consists of two 256 x 8-bit RAMs in two modes (*Table 13.3.2*). In COLR mode, the command memory stores 256 font indexes with 8-bit attributes of blinking, and sixteen colors for foreground and background; and in CCAP mode, the command memory stores 256 font indexes each with 8-bits attributes of blinking, *Italic* font option, underline font option, eight colors for foreground, and four colors for background.

Table 13.3.2 Command Memory Configuration

Register CCMODE	**Description**	**Note**
0	COLR Mode	Stores 256 font indexes with attribute for color support
1	CCAP Mode	Stores 256 font indexes with attribute for closed caption support

To generate OSD in either mode, the command memory needs to be well programmed. It is divided into title, content, and bottom sections of which initial address pointed by the registers OSDT_MADR, OSDC_MADR, and OSDB_MADR (*Figure 13.3.1*). From the initial address in each section, fill in the indexes of fonts in designated sequence, and the font will appear on the screen consecutively at the next frame. Sections allocated in the command memory can be overlaped with others.



	Addr.	Indexes from Font Memory	Attribute Setting
OSDT_MADR →	00h	15h (V)	Attribute for V
	01h	17h (X)	Attribute for X
	02h	08h (I)	Attribute for I
	03h	12h (S)	Attribute for S
	:	:	
OSDC_MADR →	15h	07h(H)	Attribute for H
	16h	1Eh (e)	Attribute for e
	17h	25h (l)	Attribute for l
	18h	25h (l)	Attribute for l
	19h	28h (o)	Attribute for o
	1Ah	43h (!)	Attribute for !
	:	:	
OSDB_MADR →	70h	0Eh (O)	Attribute for O
		0Ah (K)	Attribute for K

Figure 13.3.1 OSD Command Memory

13.4 OSD ATTRIBUTE SETTING

Each font displayed on the screen has its own 8-bit attributes for blinkings, colors, and special font formats. They are slightly different in CLOR mode and CCAP mode (*Table 13.4.1*).

Table 13.4.1 Attribute Bits Table

Bit	CLOR Mode	CCAP Mode
7	Blinking On	
6	BG Palette Color Index [2]	*Italic* Font On
5	BG Palette Color Index [1]	BG Color Index [1]
4	BG Palette Color Index [0]	BG Color Index [0]
3	FG Palette Color Index [3]	Underline Font On
2	FG Palette Color Index [2]	FG Palette Color Index [2]
1	FG Palette Color Index [1]	FG Palette Color Index [1]
0	FG Palette Color Index [0]	FG Palette Color Index [0]

There are sixteen blinking-rate options from 0.5 Hz to 7.5 Hz defined in the register, OSD_BLINK. The blinking-rate relates to the OSD_BLINK according to the following equation.

$$\text{OSD Blinking Rate} = \frac{30}{\text{OSD\_BLINK} \times 4}\text{Hz}$$

The OSD unit has sixteen build-in color palettes each can be fine adjusted from 24-bit color space (16,777,216 colors). Color palette programming can be achieved through the registers, OSD_CP_INDEX, OSD_CP_R, OSD_CP_G, and OSD_CP_B. In CLOR mode, the foreground color of each font is chosen from the color palette index 0 to 15 (FG Palette Color Index [3:0]); the background color from index 8 to 15 (BG Palette Color Index [2:0]). As in CCAP mode, the foreground color of each font is chosen from the color palette index 0 to 7 (FG Palette Color Index [2:0]); the background color of each font is chosen from the registers CCAP_BG0, CCAP_BG1, CCAP_BG2, and CCAP_BG3 with index (BG Color Index [1:0]). For CCAP_BG settings check *Table 13.4.2*.

Table 13.4.2 CCAP_BG Color Setting

CCAP_BG Bits	Color	CCAP_BG Bits	Color
0000	Black	1000	Transparent
0001	Blue	1001	Royal Blue

0010	Green	1010	Medium Aquamarine
0011	Aqua	1011	Light Green
0100	Red	1100	Orange
0101	Fuchsia	1101	Hot Pink
0110	Yellow	1110	Silver
0111	White	1111	Gray

The register, TRAN_INDEX, assigns the color palette index in which stands for transparent color.

13.5 OSD MASKING AND ALPHA-BLENDING

The VS4210's OSD unit provides special masking function in the content displaying block. This is mainly for the scrolling function during closed caption displaying. The registers, OSDC_MASK_L, OSDC_MASK_R, OSDC_MASK_T, and OSDC_MASK_B define the boundary location for the four sides of the masking blocks.

The VS4210's OSD unit also support whole screen OSD alpha-blending with the source video. The blending factor, OSD_ALPHA, is programmable in registers and follows below equation.

$$\text{OSD Displaying Color} = \frac{\text{Video Color} \times \text{OSD\_ALPHA} + \text{OSD Color} \times (4 - \text{OSD\_ALPHA})}{4}$$

13.6 OSD MEMORY ACCESS

To configure the command memory and the user-programmable font memory, the VS4210 provides two methods by accessing registers. One method is through the direct memory accessing registers OSD_ADDR, OSD_DATA, and OSD_ATRI for command memory, and registers OSD_FONT_ADDR, OSD_FONT_DATA for user-programmable font memory. The other method is through the continuous writing mechanism with Continuous-Write Registers, CW_DEST, CW_INIT_ADDR, and CW_DATA.

13.6.1 DIRECT MEMORY ACCESS

To directly change the content of the command memory, simply put the address in register OSD_ADDR and the data in registers OSD_DATA and OSD_ATRI. The OSD unit will send the data into the command memory right after the register writing operation of register OSD_DATA and OSD_ATRI. Similarily, to directly change the content of the user-programmable font memory, user simply puts the address in the register OSD_FONT_ADDR and the data in the register OSD_FONT_DATA from high byte to low byte. The OSD unit will send the data into the user-programmable font memory right after the writing operation of low byte of OSD_FONT_DATA.

13.6.2 CONTINUOUS WRITE MEMORY ACCESS

For large amount of data writing operation such as font memory writing or full screen display change, the VS4210 provides continuous writing mechanism to save host access time and efforts. To use the Continuous-Write Registers, set the destination register, CW_DEST, and the initial address register, CW_INIT_ADDR, consecutively, then write the data register, CW_DATA, repeatly. The continuous-writing mechanism will put data into the destination memory immediately after CW_DATA writing. For 16-bit data writing such as user-programmalbe font memory, the internal memory data writing will occur after every other CW_DATA writing (high byte prior to low byte). For example, to store a plus sign, “+”, into the user-programmalbe font memory, the writing sequence are as *Table 13.6.2.1*.

Table 13.6.2.1 Host Writing Sequence for Bitmap

Host Command	Address	Data
Write	CW_DEST	010
Write	CW_INIT_ADDR	00_0000_0000
Write	CW_DATA	0000_0000
Write	CW_DATA	0000_0000
Write	CW_DATA	0000_0000
Write	CW_DATA	0000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000

Host Command	Address	Data
Write	CW_DATA	0011_1111
Write	CW_DATA	1111_1100
Write	CW_DATA	0011_1111
Write	CW_DATA	1111_1100
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0001
Write	CW_DATA	1000_0000
Write	CW_DATA	0000_0000
Write	CW_DATA	0000_0000
Write	CW_DATA	0000_0000
Write	CW_DATA	0000_0000

14 PATTERN GENERATION

There is a pattern generater in the VS4210 to generate different patterns for testing or displaying.

The pattern generated is output directly to the DAC to replace the input data. The pattern generator is enabled by setting register PAT_EN(address 0xF0[7]) and different pattern modes are set by register PAT_MODE(address 0xF0[3:0]). The pattern mode is shown as Table 14.1

Table 14.1 pattern mode description

MODE	PATTERN
0	Pure Black and White video, value set by PT_RGBVAL
1	Vertical line of horizontal incremental value set by PT_INCVAL, PT_INTLEN
2	Horizontal line of vertical incremental value set by PT_INCVAL, PT_INTLEN
3	Boundary line and Cross line
4	Grid line
5	Vertical Color bar
6	Horizontal Color bar
7	Diagonal line

15 ELECTRICAL CHARACTERISTICS

15.1 ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Min	Max	Unit
Supply Voltage For Digital Core (1.8V Nominal)	V_{CCC}	-0.3	1.98	V
Supply Voltage For Digital I/O (3.3V Nominal)	V_{DDD}	-0.3	3.6	V
Supply Voltage For Analog Core (3.3V Nominal)	V_{DDA}	-0.3	3.6	V
Junction Temperature	T_J	-40	125	°C
Storage Temperature	T_{STG}	-55	125	°C
Lead Temperature (Vapor Phase Soldering, 40 Seconds)	T_L	-	215	°C
Electronic Discharge	T_{ESD}	-2000	2000	V

15.2 RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage For Digital Core (1.8V Nominal)	V_{CCC}	1.62	1.8	1.98	V
Supply Voltage For Digital I/O (3.3V Nominal)	V_{DDD}	3.0	3.3	3.6	V
Supply Voltage For Analog Core (1.8V Nominal)	V_{DDA}	1.62	1.8	1.98	V
Ambient Operation Temperature	T_A	-20	-	85	°C
Package Case Temperature	T_P	-	-	115	°C
Total Power Dissipation	P_{TOT}	-	TBA	-	W

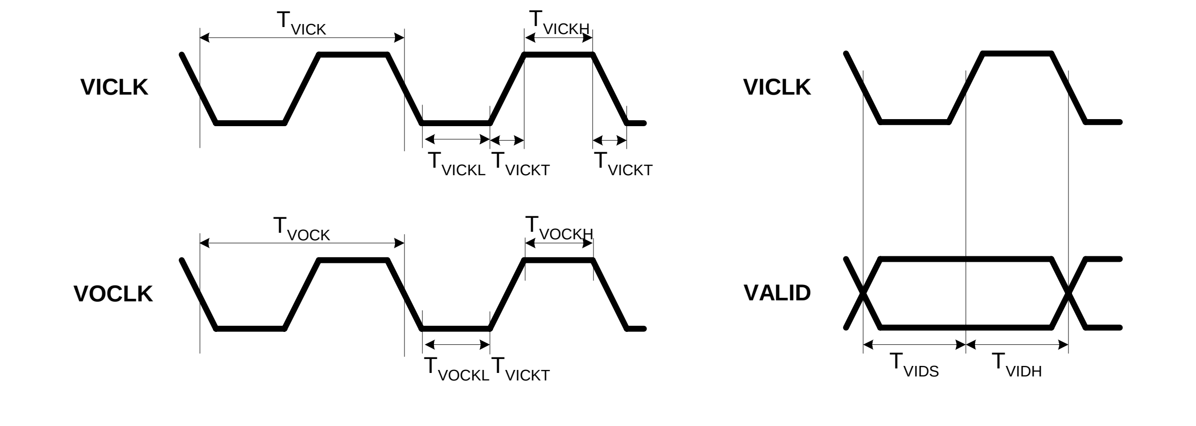
15.3 DC CHARACTERISTICS

Parameter	Symbol	Min	Typ	Max	Unit
Input (I, I_S, I_{PU}, I_{PD})					
High Level Input Voltage	V_{IH}	2	-	3.6	V
Low Level Input Voltage	V_{IL}	-0.3	-	0.8	V
Leakage Current	I_L	-	-	±1	μA
Output (O_1, O_{TS1})					
High Level Output Voltage	V_{OH}	2.4	-	-	V
Low Level Output Voltage	V_{OL}	-	-	0.4	V
Tri-State Output Leakage Current	I_L	-	-	±1	μA
Pull-Up/Down Resistor					
Pull-Up Resistor	R_{PU}	34	49	74	KΩ
Pull-Down Resistor	R_{PD}	30	47	86	KΩ

15.4 AC CHARACTERISTICS

15.4.1 VIDEO INTERFACE

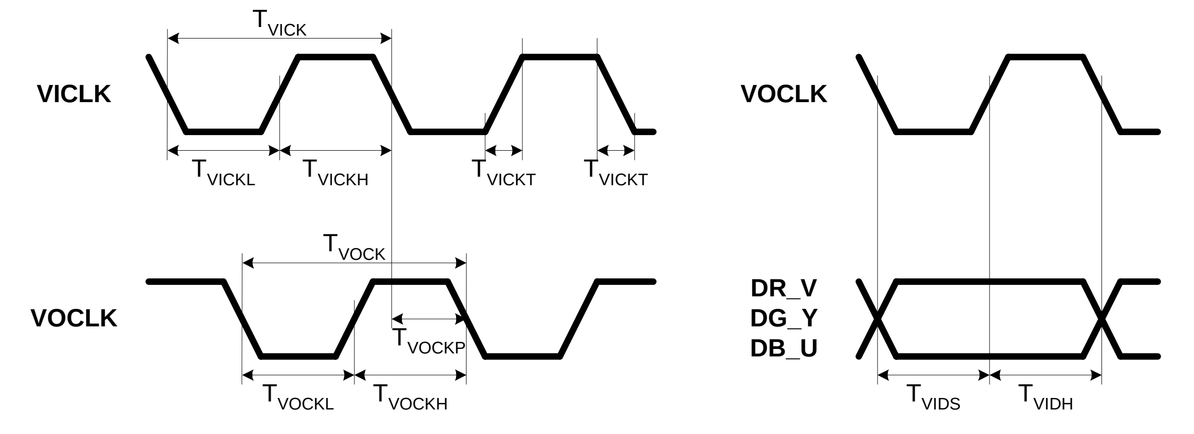
Parameter	Symbol	Min	Typ	Max	Unit
Digital Video Clocks					
Video Input Clock VICLK Frequency	F_{VICK}	-	-	150	MHz
Video Input Clock VICLK Transition Time	T_{VICKT}	-	0.5	-	ns
Video Output Clock VOCLK Frequency	F_{VOCK}	-	-	150	MHz
					ns
Digital Video Input					
Video Input Data Setup Time To VICLK	T_{VIDS}	-	0.5	-	ns
Video Input Data Hold Time From VICLK	T_{VIDH}	-	0.5	-	ns
Digital Video Output					
Video Output Data Delay From VOCLK	T_{VODH}	-	1	-	ns



Digital Video Interface Timing

15.4.2 HOST INTERFACE

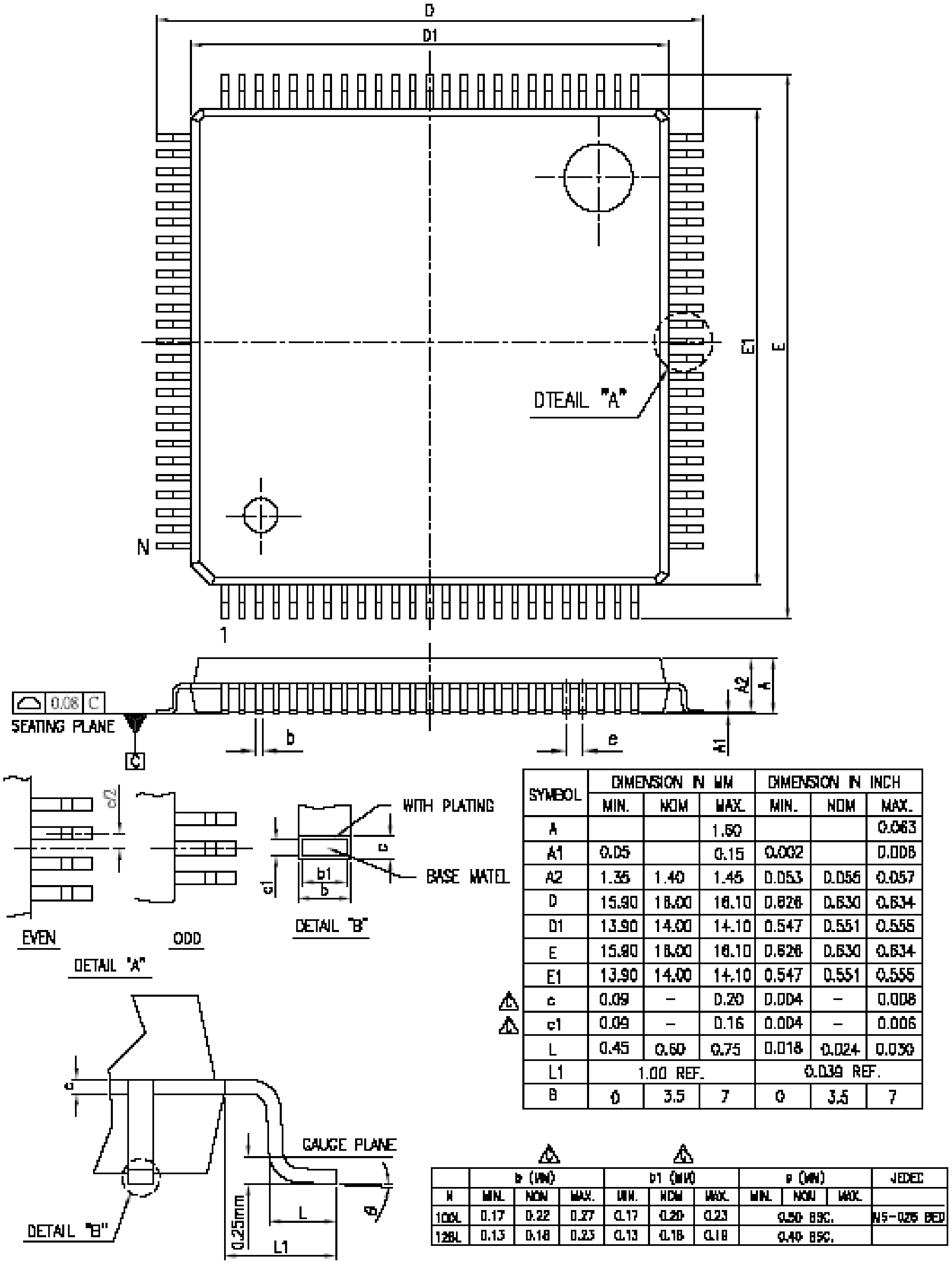
Parameter	Symbol	Min	Typ	Max	Unit
SCL Clock Frequency	F_{SCL}	-	100	-	KHz
Serial Bus Free Time Between STOP and START Condition	T_{BUF}	4.7	-	-	µs
Serial Bus Hold Time For START Condition	T_{HDSTA}	4.0	-	-	µs
SCL Clock Width Low	T_{SCLL}	4.7	-	-	µs
SCL Clock Width High	T_{SCLH}	4.0	-	-	µs
Serial Data Setup Time	T_{HSDS}	0.5	-	-	µs
Serial Data Hold Time	T_{HSDH}	-	-	0	µs
Serial Bus Setup Time For STOP Condition	T_{SUSTO}	4.0	-	-	µs



Digital Video Interface Timing

16 PACKAGE

16.1 VS4210 128PIN-LQFP



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN.	NOM	MAX.	MIN.	NOM	MAX.
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
D	15.90	16.00	16.10	0.626	0.630	0.634
D1	13.90	14.00	14.10	0.547	0.551	0.555
E	15.90	16.00	16.10	0.626	0.630	0.634
E1	13.90	14.00	14.10	0.547	0.551	0.555
⚠ c	0.09	–	0.20	0.004	–	0.008
⚠ c1	0.09	–	0.16	0.004	–	0.006
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 REF.			0.039 REF.		
θ	0	3.5	7	0	3.5	7

	⚠ b (MM)			⚠ b1 (MM)			e (MM)			JEDEC
N	MIN.	NOM	MAX.	MIN.	NOM	MAX.	MIN.	NOM	MAX.	
100L	0.17	0.22	0.27	0.17	0.20	0.23	0.50 BSC.			MS-026 BED
128L	0.13	0.18	0.23	0.13	0.16	0.18	0.40 BSC.			